

# A Software Tool for the Simulation and Optimization of Dynamic Metabolic Models

Pedro Evangelista<sup>1,2</sup>, Isabel Rocha<sup>1</sup>, Eugénio C. Ferreira<sup>1</sup>, and Miguel Rocha<sup>2</sup>

<sup>1</sup> IBB - Institute for Biotechnology and Bioengineering  
Center of Biological Engineering, University of Minho  
ptiago@deb.uminho.pt, ecferreira@deb.uminho.pt, irocha@deb.uminho.pt

<sup>2</sup> Department of Informatics / CCTC, University of Minho  
Campus de Gualtar, 4710-057 Braga, Portugal  
mrocha@di.uminho.pt

**Abstract.** In Systems Biology, there is a growing need for simulation and optimization tools for the prediction of the phenotypical behavior of microorganisms. In this paper, an open-source software platform is proposed to provide support for research in Metabolic Engineering, by implementing tools that enable the simulation and optimization of dynamic metabolic models using ordinary differential equations. Its main functionalities are related with (i) phenotype simulation of both wild type and mutant strains under given environmental conditions and (ii) strain optimization tackling tasks such as gene knockout selection or the definition of the optimal level of enzyme expression, given appropriate objective functions. The central carbon metabolism of *E. coli* was used as a case study, to illustrate the main features of the software.

## 1 Introduction

Systems Biology represents a new approach to biological research aiming to understand complex interactions at a global level, with the goal to simulate biological systems under different environments and genetic perturbations. Metabolic dynamic models allow to study the interaction of compounds in cells and its simulation results in the determination of the concentrations of the metabolites in a system along a given time interval. There are several types of dynamic models [13][18][4], but the most common is to represent metabolic networks as systems of ordinary differential equations (ODEs).

The representation of metabolism using dynamic models and its simulation has been approached before in some other computational tools: WebCell [11] is a web platform that allows to solve dynamic models and visualize system interactions; Copasi [16], that replaced its predecessor Gepasi, applies a large number of optimization techniques to dynamic model simulation to find optimum metabolite concentration (it does not try to improve the model structure, only allowing to estimate parameters based on experimental data); the BIOCHAM system [2] creates a programming environment for modeling biochemical systems, making simulations and querying the model using temporal logic; CellDesigner [10] is

a graphical visualization and modeling tool with simulation capabilities that emphasizes the creation of metabolic models using a specific graphical representation (SBGN). In this field, several standards have also been proposed, such as the Systems Biology Markup Language (SBML) [8], the standard file format for biological model representation.

On the other hand, Metabolic Engineering (ME) is a field concerned with the use of metabolic pathway modifications under an engineering perspective to attain a specific objective [17]. Since dynamic models more closely capture the properties of the metabolic network, when compared to other approaches such as stoichiometric modeling, their use on ME application would be highly desirable. However, the difficulty in obtaining kinetic data to build these models is still a major problem [14]. However, it is expected that these difficulties will be attenuated in the near future and the existence of appropriate computational tools to take full advantage of these models is of foremost importance.

The main purpose of this work is to propose a generic framework for dynamic model simulation and strain optimization, in the context of the research in ME. A novel computational framework was built including a set of user-friendly tools that should make life easier for ME researchers. The framework will mainly consist of two distinct layers: (i) phenotype simulation and (ii) strain optimization.

Regarding the simulation layer, the proposed tools enable the study of wild type strains and mutants with some gene/ reaction knockouts or over/ under-expression of some enzymes/ reactions. Furthermore, environmental conditions can be defined in both cases, by the definition of appropriate artificial reactions.

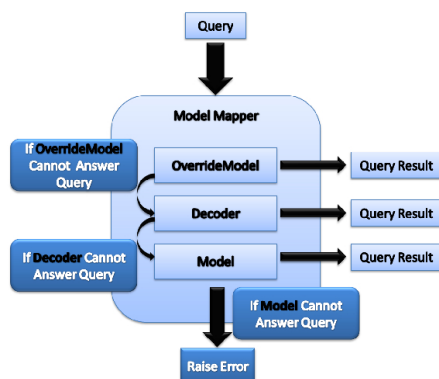
The optimization layer tackles two tasks: (i) the selection of a subset of reactions to remove from the wild type and (ii) the definition of the optimal level of expression for a set of enzymes/ reactions. In both cases, an appropriate objective function can be defined, typically related with an industrial aim. This layer is defined in a general purpose way, to allow the use of a number of optimization algorithms and to work with the simulation methods defined above. One of the major design concerns was the loose coupling between the optimization and the simulation modules to allow to optimize any model component independently of the optimization algorithm and the simulation method.

In previous work [6], the authors have used some of these tools to identify optimal or near-optimal sets of genetic changes in *E. coli* to achieve a given metabolic engineering aim using the central carbon metabolism ODE model [4]. This case study is used here to show the main capabilities of the platform.

## 2 Framework Description

### 2.1 Dynamic Model Representation

The basis for the proposed platform is a flexible representation for dynamic models that is illustrated in Figure 1. A dynamic model integrates a number of components and subsequent parameters. A unified representation is built, called *model mapper* that will answer any queries about the model components (e.g. about the model structure or parameter values). The model view is composed by



**Fig. 1.** A view of the layers involved in dynamic model representation

three layers: (3) the original model, (2) the decoder and (1) the override model. When a query is made it is passed along this (in the order 1,2,3) until one is able to answer the query. The decoder and the override model are fractional model representations. The first gives a partial model view based on a specific encoding, used mainly to provide a way to decode the solutions of possible optimization algorithms from their internal representations. The override model can be used to redefine a set of model components, thus enabling to set conditions that remain constant throughout the optimization process.

In more detail, a *model* is composed of:

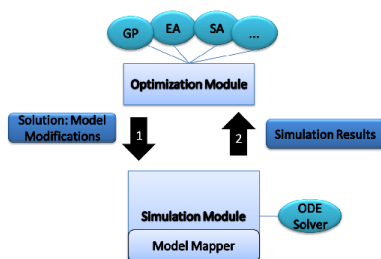
- A set of parameters, each denoted by a name and a numerical value.
- A set of variables, defined by an upper and a lower limit, an initial value and an ODE (represented by a sum of terms, where each has a multiplicative coefficient and a function).
- A set of functions, where each can be any mathematical entity that receives as its parameters the current time and a model representation, returning a numerical result. Functions can also have local parameters.

The modular architecture of the proposed framework allows to replace any component of the dynamic model. This allows the creation of multiparadigm models, since different components can use distinct modeling paradigms. This feature allows to replace less understood model components (e.g. unknown kinetic functions), for instance using models created from experimental data (e.g. a trained Neural Network).

## 2.2 Model Simulation and Optimization

The framework functionalities are divided into two logical parts: model simulation and optimization (Figure 2).

The simulation modules allow to calculate the concentrations of the metabolites in the model over a specified time interval. The simulation is based on the



**Fig. 2.** General purpose framework for dynamical model simulation and optimization

numerical integration of the ODEs in the model, specifying the time interval. This feature can be used to study the behavior of the original model (bottom layer in Figure 1) thus simulating the phenotype of the wild type strain. To simulate the phenotype of mutant strains, the override model layer is used. This allows either to remove a set of reactions from the model or to simulate the over or underexpression of an enzyme, changing the values of kinetic parameters. Also, environmental conditions can be simulated by setting the values of some external compounds to a given expression.

The optimization module allows modifications both in the model structure and in several types of parameters. The purpose is to reach model configurations that optimize a given fitness function. A user can impose changes over the model in order to simulate specific cases. Furthermore, optimization algorithms can be defined to search over the space of potential solutions, given the type of allowed changes. Fitness functions can be defined in a flexible way, i.e. no restrictions are imposed over their definition (they can be nonlinear, discontinuous, etc.). An optimization process is represented by a model, an optimization algorithm and its parameters, a decoder (to decode solutions for the optimization task) and an override model (with similar goals as the ones defined before for simulation).

Several optimization algorithms can be employed, but given the complexity of the underlying problems, the available options are meta-heuristics such as Multi-start Local Search, Simulated Annealing, several evolutionary approaches, such as Evolutionary Algorithms, Genetic Programming or Differential Evolution.

This framework currently addresses two strain optimization tasks in ME (although others might be added following the same methodology): (i) gene/ reaction knockout, i.e. to select the best subset of reactions to remove from the model; (ii) expression level, i.e., to select the optimal level of expression for a subset of enzymes (reactions) in the model. The first represents a combinatorial optimization task, while the latter can be formulated as a numerical optimization task. However, since the number of reactions within a model is typically high, the simultaneous optimization of the expression level of all reactions is generally impossible. This means that the user has to select a subset of reactions to optimize. In alternative, the set of reactions to modify and their optimal levels can be evolved simultaneously.

### 3 Software Description

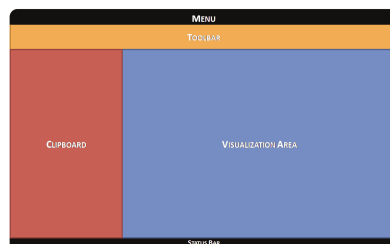
This section describes the major features involved in the development of the software platform. The application is open-source and is available in the following URL: <http://sysbio.di.uminho.pt/dynmetasim.php>. The major concerns that drove the software architecture were: (i) usability and user friendliness; (ii) portability across operating systems and architectures; (iii) compliance to standards, such as SBML; (iv) flexibility: support to a wide range of dynamic models and simulation/ optimization methods.

The whole platform was developed using the Java programming language. The graphical user interface and application logic was developed on top of AIBench[1], an environment for the development of scientific applications, based on the Model Model-View-Controller (MVC) software pattern. In AIBench, applications are built using the concepts of operations and datatypes. Operations can be mapped to an application use cases and describe the functioning and interactions of the different software components. Datatypes describe the application core types using an object-oriented approach. Specific graphic interfaces (views) are deployed to visualize certain datatypes. The main datatypes used in this platform represent the entities involved, namely models and components, simulation/ optimization setups and results.

The main functionalities of the software are the following: to load a model from an SBML file; to simulate a model given a time range and initial parameter values; to create an override model, a partial model representation that overrides part of the original model; to optimize the level of expression of a selected set of reactions; to find the best reaction knockout subset based on a given objective function; to save the results of a simulation or optimization in a file; and, to plot the simulation and optimization results.

The graphical user interface is divided into five distinct areas (Figure 3): the *Menu* that allows to launch operations; the *Toolbar* with short-cuts for some operations; the *Clipboard* area where existing objects are listed (grouped by datatypes); the *Visualization area* where the objects views are displayed; and the *Status bar* that displays information regarding current operation execution.

The software can be divided in two distinct modules: simulation engine and optimization module. In order to perform model simulation, the SUNDIALS



**Fig. 3.** Graphical user interface layout

CVODE [5] solver using *Java Native Interface* is used. CVODE is suitable for both stiff and non-stiff ODE problems and the code is available for several operating systems. The following libraries were also used: a library for optimization metaheuristics developed by the authors, *JFreeChart* [9] to display graphical simulation results and *LibSBML* [3] to parse SBML files.

## 4 Case Study

In this section, the use of the developed software is detailed in two distinct tasks, both using the *Escherichia coli* glycolysis and pentose-phosphate pathway model [4]. The model consists of mass balance equations for extra-cellular glucose and for intracellular metabolites. The mass balances take the following form:

$$\frac{dC_i}{dt} = \sum_j v_{ij}r_j - \mu C_i \quad (1)$$

where  $C_i$  represents the concentration of metabolite  $i$ ,  $\mu$  is the specific growth rate and  $v_{ij}$  is the stoichiometric coefficient for this metabolite in reaction  $j$ , the rate of which is  $r_j$ .

First of all, the simulation of the model will be shown, in the time interval  $[0,20]$  seconds using the default initial conditions (Figure 4). To run a simulation the user has to load the model. Next, she/he sets the time range for the simulation and the metabolite initial values or uses the values supplied by default in the model.

The second scenario is related with optimization, with two subtasks related with the maximization of dihydroxyacetone phosphate (DHAP): (i) Find the best knockout set; (ii) Optimize the reaction expression level by modifying the value of one of the kinetic parameters of each reaction, in this case the  $v_{max}$ . The maximization of the production of this compound was used as a case study since it has several industrial applications, including synthetic chemistry using

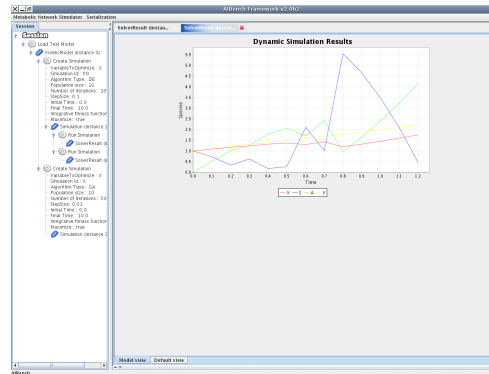


Fig. 4. A screenshot for a simulation operation

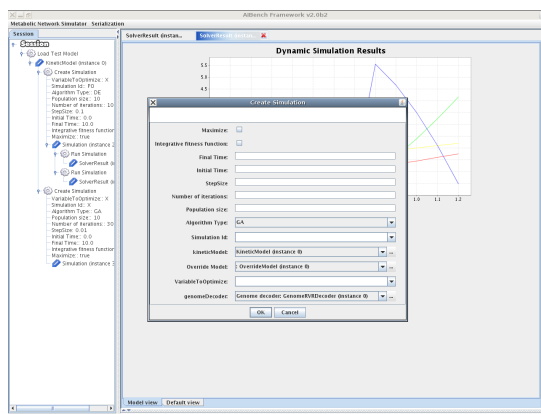


Fig. 5. Dialog for the optimization operations

the enzymatic Aldol syntheses [7]. An optimization operation can be sketched in the following steps: first, the user selects which reactions can be removed. Next, she/he selects a genome representation from the set of allowed ones. Last, she/he defines the simulation options and runs the optimization (Figure 5).

## 5 Conclusions and Further Work

In this work, a novel tool was presented that allows the representation and simulation of dynamic models and its use to achieve strain optimization tasks in Metabolic Engineering. The software package allows to simulate metabolic models based in ODEs and it is SBML compliant. Two strain optimization tasks have been addressed, namely identifying optimal sets of knockouts or expression levels for enzymes. The underlying framework for representation of dynamic models and the modular architecture of the simulation and optimization modules allow the easy extension of the platform. Furthermore, the component based approach for software development enable the easy addition of new features.

In future work, the validation of this platform with other real-world case studies will be tackled. Regarding the optimization layer, a number of other algorithms have to be integrated in the framework, namely Genetic Programming. The use of multi-objective optimization algorithms [15] is also a promising route. The integration of these tools with steady-state and constraint based computational approaches [12] will also be pursued.

## References

1. AIBench, <http://www.aibench.org/>
2. BioCham, <http://contraintes.inria.fr/BIOCHAM/>
3. Bornstein, B., Keating, S.M., Jouraku, A., Hucka, M.: LibSBML: an API Library for SBML. *Bioinformatics* 24(6), 880–881 (2008)

4. Chassagnole, C., Noisommit-Rizzi, N., Schmid, J.W., Mauch, K., Reuss, M.: Dynamic modeling of the central carbon metabolism of *Escherichia coli*. *Biotechnology and Bioengineering* 79(1), 53–73 (2002)
5. Cohen, S., Hindmarsh, C.: Cvode, a stiff/nonstiff ode solver in c. *Computers in Physics* 10(2), 138–143 (1996)
6. Evangelista, P., Rocha, M., Rocha, I., Ferreira, E.C.: Evolutionary approaches for strain optimization using dynamic models under a metabolic engineering perspective. In: *Proc. of the EvoBio 2009* (to appear) (April 2009)
7. Gefflaut, T., Lemaire, M., Valentin, M., Bolte, J.: A novel efficient synthesis of dihydroxyacetone phosphate and bromoacetol phosphate for use in enzymatic aldol syntheses. *The Journal of Organic Chemistry* 62(17), 5920–5922 (1997)
8. Hucka, M., Finney, A., et al.: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19(4), 524–531 (2003)
9. JFreeChart, <http://www.jfreechart.org/jfreechart>
10. Kitano, H.: Systems biology: a brief overview. *Science* 295(5560), 1662–1664 (2002)
11. Lee, D.Y., Yun, C., Hou, B., Park, S., Lee, S.Y.: Webcell: a web-based environment for kinetic modeling and dynamic simulation of cellular networks. *Bioinformatics* 22(9), 1150–1151 (2006)
12. Rocha, M., Maia, P., Mendes, R., Ferreira, E.C., Patil, K., Nielsen, J., Rocha, I.: Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC Bioinformatics* 9(499) (2008)
13. Nummela, J., Julstrom, B.A.: Evolving petri nets to represent metabolic pathways. In: *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 2133–2139. ACM, New York (2005)
14. Patil, K., Akessin, M., Nielsen, J.: Use of genome-scale microbial models for metabolic engineering. *Current Opinion Biotechnology* 15(1) (2004)
15. Maia, P., Ferreira, E.C., Rocha, I., Rocha, M.: Evaluating evolutionary multiobjective algorithms for the in silico optimization of mutant strains. In: *Proc. IEEE Intern. Conf. BioInformatics and BioEngineering (BIBE 2008)*, Athens (2008)
16. Sahle, S., Gauges, R., et al.: Simulation of biochemical networks using copasi: a complex pathway simulator. In: *WSC 2006: Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference*, pp. 1698–1706 (2006)
17. Stephanopoulos, G.: Metabolic fluxes and metabolic engineering. *Metabolic Engineering* 1(1), 1–11 (1999)
18. Yang, K., Ma, W., Liang, H., Ouyang, Q., Tang, C., Lai, L.: Dynamic simulations on the arachidonic acid metabolic network. *PLoS Computational Biology*, e55.eor+ (February 2007)